

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-295694

(43)Date of publication of application : 10.11.1995

(51)Int.Cl.

G06F 1/32
G01C 15/00
G06F 9/46

(21)Application number : 06-086669

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 25.04.1994

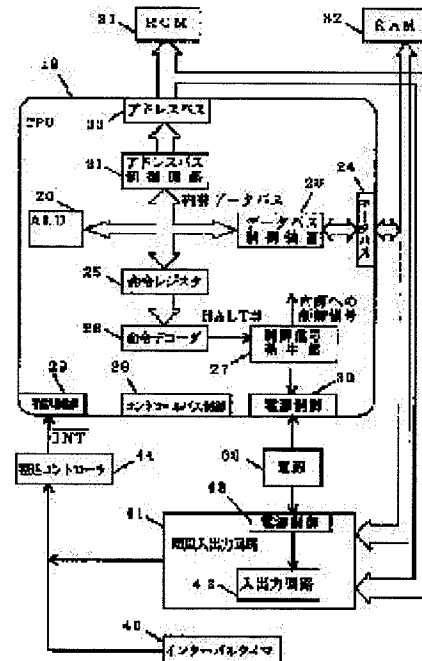
(72)Inventor : IDE MITSUNORI

(54) POWER SAVING METHOD FOR ARITHMETIC PROCESSOR

(57)Abstract:

PURPOSE: To extend a battery actuation time by reducing the power consumption when a multitask OS is used.

CONSTITUTION: A conventional method executes an idle task when there is no task to be run, and this idle task becomes an endless loop of NOP instructions, but this method replaces the endless loop of NOP instructions with an instruction (e.g. HALT instruction) for placing a CPU 19 in power-saving mode, thereby reducing the power consumption. In the power-saving mode, the CPU 19 stops operating and the power source of a device except a part which is required to reset the mode is stopped by the CPU itself. The CPU exits from the power-saving mode by external interruption.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-295694

(43) 公開日 平成7年(1995)11月10日

(51) Int.Cl. ⁵	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 1/32				
G 0 1 C 15/00				
G 0 6 F 9/46	3 4 0 B	7737-5B		

G 0 6 F 1/ 00 3 3 2 Z

審査請求 未請求 請求項の数 3 O L (全 4 頁)

(21) 出願番号 特願平6-86669

(22) 出願日 平成6年(1994)4月25日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 井手 光則

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

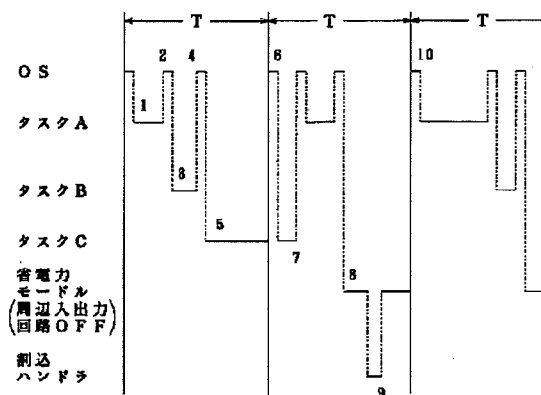
(74) 代理人 弁理士 小鍛冶 明 (外2名)

(54) 【発明の名称】 演算処理装置の省電力方法

(57) 【要約】

【目的】 マルチタスクOS使用時において、低消費電力を実現し、バッテリー起動時間を延長することを目的とする。

【構成】 従来方法では、走るべきタスクがない時はアイドルタスクが実行され、このアイドルタスクはNOP命令の無限ループとなるが、本発明においては、NOP命令の無限ループをCPU19を省電力モードにする命令(例えばHALT命令)に置き換えることにより、低消費電力を実現する(8)。省電力モードにおいてはCPU19はその動作を停止しており、モードの解除に必要な部分以外の装置の電源をCPU自身が停止している。省電力モードからの脱出は外部割り込みによって行なわれる。



【特許請求の範囲】

【請求項1】省電力モードを持つ演算処理装置と、前記演算処理装置に定期的に割込をかけるインターバルタイマ部と、前記演算処理装置からの命令に従うかまたは自ら周辺装置との入出力の制御を行ないデータ収集や演算処理装置に割込をかける周辺入出力回路と、上記インターバルタイマ部と前記周辺入出力回路からの割込をその優先順位等で制御する割込コントローラ部を持つ装置において、

上記インターバルタイマ部による割込に従いマルチタスクが実現されている場合において、走るべきタスクがない時に演算処理装置を省電力モードにする事を特徴とする演算処理装置の省電力方法。

【請求項2】省電力モードを持っていない演算処理装置と、前記演算処理装置に定期的に割込をかけるインターバルタイマ部と、前記演算処理装置からの命令に従うかまたは自ら周辺装置との入出力の制御を行ないデータ収集や演算処理装置に割込をかける周辺入出力回路と、上記インターバルタイマ部と前記周辺入出力回路からの割込をその優先順位等で制御する割込コントローラ部を持つ装置において、上記インターバルタイマ部による割込に従いマルチタスクが実現されている場合において、走るべきタスクがない時に不必要な周辺入出力回路の電源を切る事を特徴とする演算処理装置の省電力方法。

【請求項3】省電力モードを持つ演算処理装置において、演算処理装置を低消費電力モードにし、さらに、周辺入出力装置の電源を落とす事を特徴とする請求項1若しくは請求項2記載の演算処理装置の省電力方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、演算処理装置の省電力方法に関するものである。

【0002】

【従来の技術】演算処理装置（以下、CPUとする）を動作させるOSとしてマルチタスクOSを使用した時、何も動くタスクがなかったら（アイドル状態）、通常OSはアイドルタスクを起動する。これは一般的に、図6に示すように何も処理しない命令（例えば、NOP命令）の無限ループで構成されている。

【0003】アイドルタスクの実行は次の周期割込におけるスケジューリングまで継続実行はされない。すなわちアイドルタスクはアイドル状態のみスケジューリングされるタスクである。

【0004】一般的に、アイドルタスクに限らず、普通のタスクは割込許可状態で走っている。このため、タスク起動中に（周期割込以外の）割込が発生すると、割り込み要因に対応した割込処理プログラムが実行され、終了後元のタスクに戻る。

【0005】

【発明が解決しようとする課題】このように従来のマル

チタスクOSではアイドル状態の時、アイドルタスクが起動されており、無駄に電力を消費していた。

【0006】本発明は、マルチタスクOS下において省電力を可能にする演算処理装置の省電力方法を提供する事を目的とする。

【0007】

【課題を解決するための手段】本発明は上記問題点を解決するため、省電力モードを持つCPUと、CPUに定期的に割込をかけるインターバルタイマ部と、CPUからの命令に従うかまたは自ら周辺装置との入出力の制御を行ないデータ収集やCPUに割込をかける周辺入出力回路と、インターバルタイマ部と周辺入出力回路からの割込をその優先順位等で制御する割込コントローラ部を持つ装置において、インターバルタイマ部による割込に従いマルチタスクが実現されている場合において、走るべきタスクがない時にCPUを省電力モードにする。

【0008】

【作用】本発明は上記構成により、マルチタスクOS下のアイドル状態において、CPUを省電力モードにして省電力を実現する事が出来る。

【0009】

【実施例】以下、本発明の一実施例を説明する。図2はCPU（演算処理装置）の内部構成およびCPUを用いたマルチタスクOS実現のための最小ハードウェア構成図である。19はCPUである。以下、CPU19の構成について説明する。20は数値演算や論理演算を行なうALU、22はアドレスバスであり、21はアドレスバスを制御するアドレスバス制御回路である。また24はデータバスであり、23はデータバスを制御するデータバス制御回路である。データバス24から読込まれた命令は命令レジスタ25を介して、命令デコーダ26に入力され、解析された結果に従い、制御信号発生部27に命令を出すことにより、いろいろな内部制御を行なう。28は外部割込のための外部割込制御回路であり、29はWRやRD信号等を制御するコントロールバス制御回路、30は電源制御回路である。以上がCPU19の構成である。

【0010】31はROM、32はRAM、33は外部電源である。40は、定期的に（図1ではT）割込をかけるためのインターバルタイマである。41は入出力回路42と電源制御回路43からなる周辺入出力回路である。以下、周辺入出力回路41について説明する。

【0011】周辺入出力回路41はCPU19とはアドレスバスとデータバスでつながっており、周辺回路はこれらのバスを介して制御される。また周辺入出力回路41は受信割込等の割込を発生できるようになっており、その割込出力端子はインターバルタイマ40と共に割込コントローラ44に入力される。割込コントローラ44は、各種割込の優先順位等に従い、CPU19に割込をかける。その割込要因は、割込発生時にデータバス24

に乗せられている。

【0012】OSは割込を検知すると、その割込要因を調べ、その割込が周期割込の場合には、タスクの切替処理（スケジューリング）を行なう。スケジューリング処理で参照されるデータは、あらかじめ登録されたタスクの起動順を格納したメモリである（これを待ち行列という）。このスケジューリング処理は、周期割り込みだけでなく、タスク自らが実行権を放棄した場合にも行われる。

【0013】図1は本実施例における省電力方法のタスクスケジューリングを示す説明図である。以下、図に基づいて具体的に説明する。

【0014】まず、OSはタスク自らの実行権放棄もしくは周期割り込みにより、タスクをスケジューリングする。一例として、OSの管理下に3つのタスクA、B、Cがあるとすると、Tはインターバルタイマからの周期割込の周期である。またこの例では、最初の周期1ではアイドル状態にならなかったとする。

【0015】今、タスクAが走っているとすると（1）。ここで、タスクAが実行権の放棄をするためOSのモニターコールをすると、OSに制御が移り（2）、次に動くタスクBを起動する（3）。これをスケジューリングという。すなわち、スケジューリングとは、待ち行列からタスクAの後に動くタスクを検索し、そのタスクを起動させる処理である。

【0016】同様に、タスクBが実行権を放棄すると、OSによりスケジューリングが行なわれて（4）、タスクCが起動される（5）。

【0017】もし、タスクCが周期割込の周期Tまでに実行権を放棄しなくても、周期割込が発生すると、OSに制御が移り（6）、スケジューリングが行なわれる。この場合、タスクCの実行中の周期割込であるので、次の周期2ではタスクCが継続実行される（7）。

【0018】周期2にて、タスクCの実行権放棄後、前回のタイミングと同様の手順でタスクAが起動される。ここでは、タスクAの実行権放棄後に走るタスクがない（すなわちアイドル状態）とする。

【0019】従来の方法ではこの後アイドルタスクが実行され、このアイドルタスクは図6のようにNOP命令の無限ループとなるが、本発明の実施例においては、図3のようにNOP命令の無限ループをCPU19を省電力モードにする命令（例えばHALT命令）に置き換えることにより、低消費電力を実現する（8）。省電力モードにおいてはCPU19はその動作を停止しており、モードの解除に必要な部分以外の装置の電源をCPU自身が停止している。省電力モードからの脱出は外部割り込みによって行なわれる。省電力モード時の割り込みに対するレスポンスと、非省電力モード時の割り込みに対するレスポンスは若干違う場合もあるが、ほとんど無視できる。

【0020】また、図2における周辺入出力回路41は、CPU19と、アドレスバスとデータバスとでつながっており、周辺回路の電源OFFはこれらのバスを介した命令発行で行うことが出来るようになっている。もし、CPU19が省電力モードを備えていなかったら、NOP命令をHALT命令に置き換えることは出来ないが、その代りに、図4のように無限ループに入る前に不必要な周辺入出力回路の電源をOFFにすることにより低消費電力を実現する。

【0021】さらに、図5のようにCPU19を省電力モードにし周辺回路の電源を落とすことにより、いっそうの低消費電力化を実現できる。

【0022】CPU19の省電力モード中は、一般に割り込み許可であるので、符号4が示すように割り込み処理プログラムを実行することは可能であり、処理の抜けが生じることはない。省電力モード中に割り込みが生じると、省電力モードから脱出し、OSによるスケジューリングが行なわれる。

【0023】

【発明の効果】以上のように本発明は、インターバルタイマ部による割込に従いマルチタスクが実現されている場合において、走るべきタスクがない時にCPUを省電力モードにすることにより、マルチタスクOS使用の装置において省電力を実現でき、特にバッテリー駆動の装置においては有用な発明である。

【図面の簡単な説明】

【図1】本発明の一実施例における省電力方法のタスクスケジューリングを示す説明図

【図2】本発明の一実施例において演算処理装置を用いたマルチタスクOS実現のための最小ハードウェア構成図

【図3】本発明の一実施例における省電力方法のフローチャート

【図4】本発明の一実施例における省電力方法のフローチャート

【図5】本発明の一実施例における省電力方法のフローチャート

【図6】従来のマルチタスクOSを使用した装置のアイドル状態のフローチャート

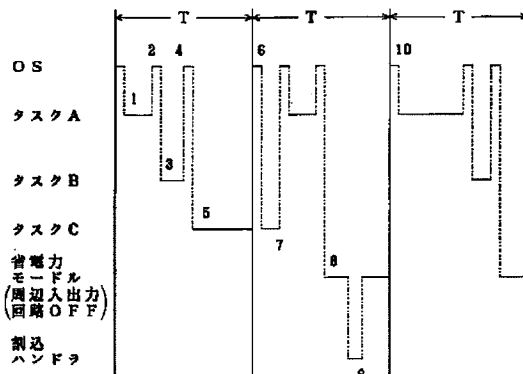
【符号の説明】

- 8 省電力モードの実行
- 9 省電力モードにおける割込ハンドラの起動
- 19 CPU
- 20 ALU
- 21 アドレスバス制御回路
- 22 アドレスバス
- 23 データバス制御回路
- 24 データバス
- 25 命令レジスタ
- 26 命令デコーダ

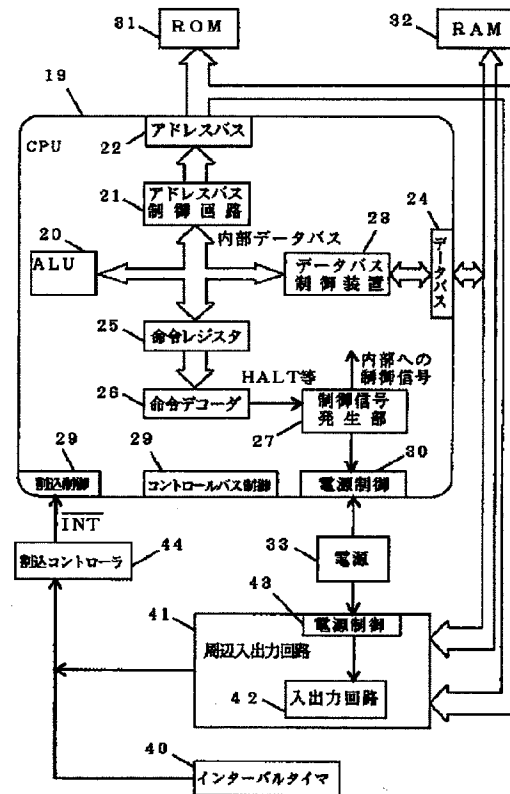
27 制御信号発生部
28 外部割込制御回路
29 コントロールバス制御回路
30 電源制御回路
31 ROM
32 RAM

* 33 外部電源
40 インターバルタイマ
41 周辺入出力回路
42 入出力回路
43 電源制御
* 44 割込コントローラ

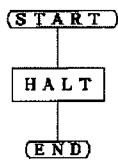
【図1】



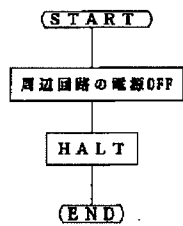
【図2】



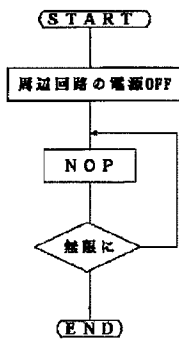
【図3】



【図4】



【図5】



【図6】

